

Abstract of JP2000-57081

PROBLEM TO BE SOLVED: To enable a user to select and connect his desired one of peripheral devices of different systems by selecting the continuation of execution or the control shift of an application program according to the system of the peripheral device which is connected to a device interface.

SOLUTION: A CPU 1 decides whether or not the display device, which is connected to a display device interface 12, is equal to a 1st display device X of a segment system corresponding to an application program A based on the display device information stored in a data RAM 4. If the display device information shows X, the connection of the display device X is decided and the execution of the program A is carried on. If the display device information shows Y, the connection of a 2nd display device Y is decided. Then the start is called for an application program B of a dynamic link library form corresponding to the display device Y. Thereafter, the control is shifted to the program B.

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2000-57081

(P2000-57081A)

(43)公開日 平成12年2月25日 (2000.2.25)

(51)Int.Cl.

G 0 6 F 13/10
9/06

識別記号

3 2 0
4 1 0

F I

G 0 6 F 13/10
9/06

コード*(参考)

3 2 0 A 5 B 0 1 4
4 1 0 S 5 B 0 7 6

審査請求 未請求 請求項の数1 O L (全 6 頁)

(21)出願番号 特願平10-228948

(22)出願日 平成10年8月13日 (1998.8.13)

(71)出願人 000003562

東芝テック株式会社

東京都千代田区神田錦町1丁目1番地

(72)発明者 小川 昌秀

静岡県田方郡大仁町大仁570番地 株式会
社テック大仁事業所内

(74)代理人 100058479

弁理士 鈴江 武彦 (外6名)

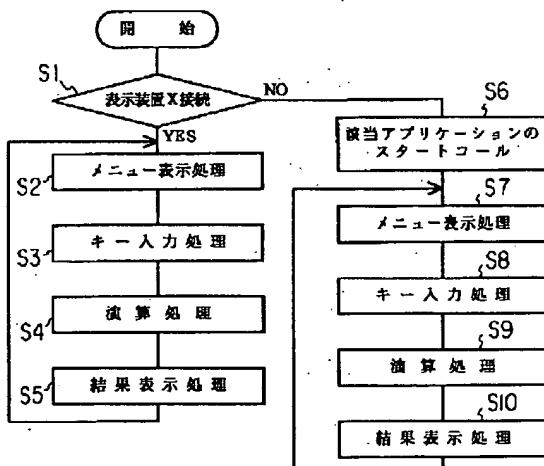
F ターム(参考) 5B014 EB01 FA06 FB04
5B076 AB04

(54)【発明の名称】 情報処理装置

(57)【要約】

【課題】異なる方式の周辺装置の中からユーザが所望する周辺装置を選択して接続できるようにする。

【解決手段】周辺装置を使用する業務のアプリケーションプログラムとして周辺装置の各方式別のアプリケーションプログラムを作成し、そのうちの1つを実行形式として実装し、他のアプリケーションプログラムをダイナミックリンクライブラリ形式として実装する。そして、実行形式のアプリケーションプログラムが起動したことに応じてデバイスインターフェースに接続された周辺装置の方式を調べ、その方式が実行形式のアプリケーションプログラムに対応する場合は当該アプリケーションプログラムの実行を継続し、他のアプリケーションプログラムに対応する場合はその対応するアプリケーションプログラムを起動させて以後はその対応するアプリケーションプログラムに制御を移す。



【特許請求の範囲】

【請求項1】 デバイスインターフェースに表示装置等の周辺装置として方式の異なる周辺装置を選択的に接続可能なした情報処理装置において、前記周辺装置を使用する業務のアプリケーションプログラムとして前記周辺装置の各方式別のアプリケーションプログラムを作成し、そのうちの1つをオペレーティングシステム立上げ後に起動する実行形式として実装し、他のアプリケーションプログラムを前記実行形式のアプリケーションプログラムを実行している際の必要なときに起動するダイナミックリンクライブラリ形式として実装するとともに、前記実行形式のアプリケーションプログラムが起動したことに対応して前記デバイスインターフェースに接続された周辺装置の方式を調べ、その方式が前記実行形式のアプリケーションプログラムに対応する場合は当該アプリケーションプログラムの実行を継続し、他のアプリケーションプログラムに対応する場合はその対応するアプリケーションプログラムを起動させて以後はその対応するアプリケーションプログラムに制御を移すことを特徴とする情報処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、表示装置などの周辺装置として方式の異なる周辺装置を選択的に接続可能な情報処理装置に関する。

【0002】

【従来の技術】従来の情報処理装置、例えば周知のパソコン用コンピュータに、POS (Point Of Sales) 専用のキーボード、顧客用の表示装置、ドロワ、レシート/ジャーナルプリンタなどの周辺装置を装着することによって、当該パソコン用コンピュータをPOSターミナルとして機能させることは可能である。また、この場合において、例えば表示装置としてはセグメント表示方式のものやドット表示方式のものなどがあるが、ユーザが使用目的に応じて所望する方式の表示装置を適宜選択できることが好ましい。そのためには、例えば表示装置としてセグメント表示方式のものが接続された場合でも、またドット表示方式のものが接続された場合でも同様に機能するように、表示装置を使用する各種業務のアプリケーションプログラムを作成しなければならない。

【0003】 しかし従来は、このようなアプリケーションプログラムが開発されていなかったので、周辺装置として方式が異なる複数種類が実存していても、使用可能なもののはいずれか1種類に制限されているのが実情であった。

【0004】 そこで、このようなアプリケーションプログラムの作成方法を考察してみると、第1の方法として、各方式の周辺装置にそれぞれ対応した複数のアプリケーションプログラムを用意し、これらのアプリケーションプログラムを、いずれもオペレーティングシステム

立上げ後に起動する実行形式としてパソコン用コンピュータに実装する方法が考えられる。この方法であれば、周辺装置としてどのような方式の周辺装置が接続されているかは、オペレーティングシステムの立上げ時に判別されるので、接続された周辺装置の方式に対応したアプリケーションプログラムは、起動後、オペレーティングシステムの制御下で処理を実行するようになり、他の方式に対応したアプリケーションプログラムは永久にスリープするようになる。その結果、どのような方式の周辺装置が接続された場合でも同様に機能するようになる。

【0005】 また第2の方法として、方式が異なる周辺装置にそれぞれ対応した各アプリケーションプログラムを結合して1つのアプリケーションプログラムを作成し、このアプリケーションプログラムを実行形式としてパソコン用コンピュータに実装する方法も考えられる。この場合、周辺装置を使用する処理の開始前に接続されている周辺装置の種類を判別しその判別結果に応じたデータの編集や周辺装置の動作指令を行うようにアプリケーションプログラムを作成する必要がある。

【0006】

【発明が解決しようとする課題】しかしながら、アプリケーションプログラムを実行形式としてパソコン用コンピュータに実装する場合には、そのアプリケーションプログラムに固有のタスクIDコードを付すことによって他のアプリケーションプログラムと識別化する必要がある。このため、第1の方法では、各アプリケーションプログラムにそれぞれ異なるタスクIDコードを付さなければならず、タスクIDコードが増加する懸念があった。一方、第2の方法では、周辺装置をいずれか1種類の方式のものに制限した既存のものとタスクIDコードの数は変わらないが、アプリケーションプログラムの作成に大きな手間と時間を要する問題があった。

【0007】 本発明はこのような事情に基づいてなされたもので、その目的とするところは、タスクIDコードが増加せず、しかもアプリケーションプログラムを作成し直す面倒無しに、異なる方式の周辺装置の中からユーザが所望する周辺装置を選択して接続できる情報処理装置を提供しようとするものである。

【0008】

【課題を解決するための手段】本発明は、デバイスインターフェースに表示装置等の周辺装置として方式の異なる周辺装置を選択的に接続可能にした情報処理装置において、周辺装置を使用する業務のアプリケーションプログラムとして周辺装置の各方式別のアプリケーションプログラムを作成し、そのうちの1つをオペレーティングシステム立上げ後に起動する実行形式として実装し、他のアプリケーションプログラムを前記実行形式のアプリケーションプログラムを実行している際の必要なときに起動するダイナミックリンクライブラリ形式として実装するとともに、実行形式のアプリケーションプログラムが

起動したことに応じてデバイスインターフェースに接続された周辺装置の方式を調べ、その方式が実行形式のアプリケーションプログラムに対応する場合は当該アプリケーションプログラムの実行を継続し、他のアプリケーションプログラムに対応する場合はその対応するアプリケーションプログラムを起動させて以後はその対応するアプリケーションプログラムに制御を移すようにしたものである。

【0009】

【発明の実施の形態】以下、本発明の一実施の形態を図面を用いて説明する。図1はこの一実施の形態の情報処理装置の要部を示すブロック図であって、この情報処理装置は、制御部本体としてCPU (Central Processing Unit) 1を搭載している。また、記憶部としてブートプログラムなどの固定的データを予め格納するROM (Read Only Memory) 2、オペレーティングシステムや各種業務に対応したアプリケーションプログラムなどのプログラムデータを記憶するプログラムRAM (Random Access Memory) 3、及び設定データや情報処理データなどを一時格納するデータRAM 4を搭載している。そしてCPU 1と記憶部2, 3, 4とを、アドレスバス、データバス等のバスライン5で接続している。

【0010】またこの情報処理装置は、補助記憶装置としてHDD (Hard Disc Drive) 装置6とFDD (Floppy Disk Drive) 装置7を搭載しており、これらの補助記憶装置6, 7を、それぞれコントローラ8, 9を介して前記バスライン5に接続している。さらにこの情報処理装置は、前記バスライン5に、回線を通じてデータ伝送を行う通信インターフェース10の他、デバイスインターフェースとしてキーボードインターフェース11や表示装置インターフェース12などを接続しており、これらのデバイスインターフェース11, 12にそれぞれキーボード、表示装置などの周辺装置（入出力装置）を接続できるようになっている。特に、本実施の形態では、表示装置として、セグメント表示方式を採用している第1の表示装置Xまたはドット表示方式を採用している第2の表示装置Yのいずれか一方をユーザが任意に選択して接続できるようになっている。

【0011】図2はかかる情報処理装置が実装する主要なプログラム構造を示す図である。図示するようにこの情報処理装置は、基本プログラムとしてのオペレーティングシステム21及び入出力装置に依存した処理を行うデバイスドライバ22の他、各種業務に対応したアプリケーションプログラムA, B, C, D, …を実装している。

【0012】ここで、アプリケーションプログラムAとアプリケーションプログラムBは、いずれも表示装置を使用する同一業務のアプリケーションプログラムであり、その構成は、図3に示すように、メニュー表示処理31、キー入力処理32、演算処理33及び結果表示処

理34をその順番に繰返し実行するオブジェクト構成である。そして、アプリケーションプログラムAとアプリケーションプログラムBとは、処理名（関数名）を同一にし、タスクIDコードも共通にしている。ただし、アプリケーションプログラムAのメニュー表示処理31及び結果表示処理34は、セグメント表示方式の第1の表示装置Xに対応して作成しており、アプリケーションプログラムBのメニュー表示処理31及び結果表示処理34は、ドット表示方式の第2の表示装置Yに対応して作成している。そして、アプリケーションプログラムAは、従来と同様、オペレーティングシステム21の立上げ後に起動する実行形式として実装しており、アプリケーションプログラムBは、上記実行形式のアプリケーションプログラムAを実行している際の必要なときに起動するダイナミックリンクライブラリ形式として実装している。

【0013】さて、このように構成された本実施の形態の情報処理装置において、電源が投入されてオペレーティングシステム21の立上げ処理が開始されると、図4に示すように、その中でCPU 1は、表示装置インターフェース12に表示装置が接続されているか否かをチェック（S1）、接続されている場合にはその接続機種がセグメント表示方式の第1の表示装置Xなのか、ドット表示方式の第2の表示装置Yなのかを判別する（S2）。そして、第1の表示装置Xの場合には表示装置情報XをデータRAM 4に記憶し（S3）、第2の表示装置Yの場合には表示装置情報YをデータRAM 4に記憶する（S4）。なお、表示装置インターフェース12に表示装置が接続されていない場合には、表示装置のみ接続エラーとして、エラー処理ルーチンに入る。

【0014】こうして、オペレーティングシステム21が立ち上がるとき、このオペレーティングシステム21の制御下で実行形式の各種アプリケーションプログラムが起動可能となる。この状態で、前記アプリケーションプログラムA及びアプリケーションプログラムBで処理する業務の処理名がOutputされ該当するタスクIDコードがコールされると、実行形式であるアプリケーションプログラムAが起動する。

【0015】これにより、CPU 1は、図5に示すように、先ず表示装置インターフェース12に接続されている表示装置がアプリケーションプログラムAに対応するセグメント表示方式の第1の表示装置Xであるか否かをデータRAM 4に記憶している表示装置情報から判断する（S1）。ここで表示装置情報がXの場合には、第1の表示装置Xが接続されていると判断し、このアプリケーションプログラムAの実行を継続する。そして、このアプリケーションプログラムAに従い、メニュー表示処理（S2）、キー入力処理（S3）、演算処理（S4）及び結果表示処理（S5）をこの順番に繰返し実行する。

【0016】これに対し、表示装置情報がYの場合に

は、第2の表示装置Yが接続されていると判断し、この第2の表示装置Yに対応したダイナミックリンクライブラリ形式のアプリケーションプログラムBをスタートコードする(S6)。そしてそれ以後は、このアプリケーションプログラムBに制御を移し、メニュー表示処理(S7)、キー入力処理(S8)、演算処理(S9)及び結果表示処理(S10)をこの順番に繰返し実行する。

【0017】したがって、表示装置インターフェース12にセグメント表示方式の第1の表示装置Xが接続された場合には、この第1の表示装置Xに対応したメニュー表示処理31及び結果表示処理34を備えたアプリケーションプログラムAが実行されるので、メニュー表示及び結果表示が第1の表示装置Xに対して確実に行われる。また、表示装置インターフェース12にドット表示方式の第2の表示装置Yが接続された場合には、この第2の表示装置Yに対応したメニュー表示処理31及び結果表示処理34を備えたアプリケーションプログラムBが実行されるので、メニュー表示及び結果表示が第2の表示装置Yに対して確実に行われる。

【0018】このように本実施の形態の情報処理装置であれば、表示方式が異なる第1の表示装置Xと第2の表示装置Yのどちらを接続しても正常に動作させることができるので、異なる方式の表示装置X、Yの中からユーザーが所望する表示装置を選択して接続できるようになる。この場合において、第1の表示装置Xに対応した表示処理を有するアプリケーションプログラムAを実行形式で実装し、第2の表示装置Yに対応した表示処理を有するアプリケーションプログラムBを上記アプリケーションプログラムAを実行している際の必要なときに起動するダイナミックリンクライブラリ形式で実装しているので、同一業務を実行するアプリケーションプログラムAとアプリケーションプログラムBのタスクIDコードを共通に設定できる。したがって、同一業務を実行するアプリケーションプログラムを接続可能な表示装置の数

だけ実装しても、タスクIDコードが増加する懸念はなくなる。また、アプリケーションプログラムAとアプリケーションプログラムBとを結合させる必要もないことで、アプリケーションプログラムを作成し直す面倒もない。

【0019】なお、前記一実施の形態では、表示方式の異なる表示装置を選択的に接続可能な情報処理装置に本発明を適用した場合を示したが、例えば印字方式（シリアル印字方式とライン印字方式など）の異なるプリンタを接続可能な情報処理装置に対しても本発明を同様に適用できるものである。

【0020】

【発明の効果】以上詳述したように本発明によれば、タスクIDコードが増加せず、しかもアプリケーションプログラムを作成し直す面倒無しに、異なる方式の周辺装置の中からユーザーが所望する周辺装置を選択して接続できる情報処理装置を提供できる。

【図面の簡単な説明】

【図1】 本発明の一実施の形態である情報処理装置の要部構成を示すブロック図。

【図2】 同実施の形態の情報処理装置に実装される主要なプログラム構造を示す図。

【図3】 同実施の形態におけるアプリケーションプログラムA、Bの構成を示す図。

【図4】 同実施の形態におけるオペレーティングシステムの立ち上げ処理の要部を示す流れ図。

【図5】 同実施の形態におけるアプリケーションプログラムA、Bの処理の要部を示す流れ図。

【符号の説明】

1…CPU

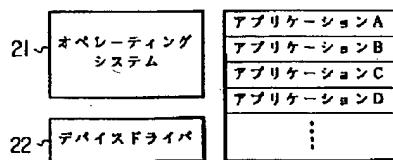
3…プログラムRAM

12…表示装置インターフェース

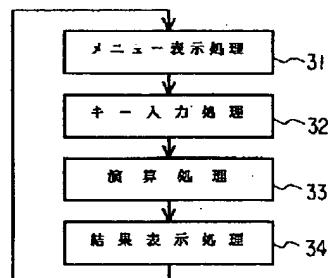
X…第1の表示装置

Y…第2の表示装置

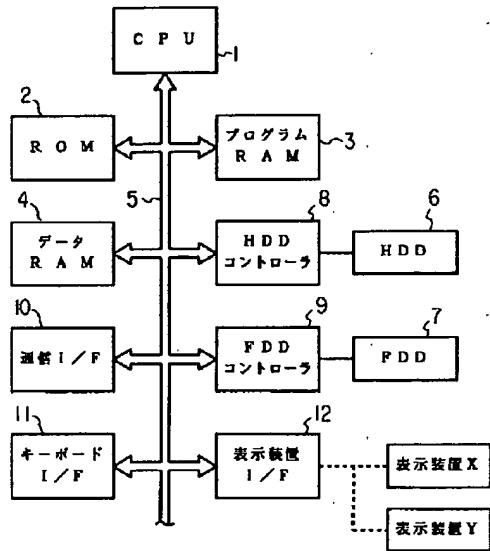
【図2】



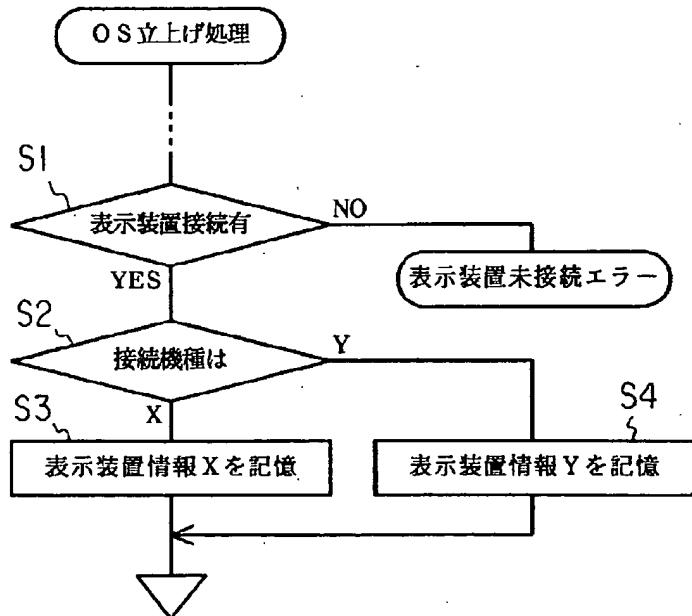
【図3】



【図1】



【図4】



【図5】

